

功能：对数据点实现参数化

输入参数：m_aVertex-数据点；m_pType-参数化类型，由数据点参数化对话框输入；两者均为受保护成员。

输出参数：m_aU-数据点参数值数组，受保护成员。

调用函数：Normalize()-对参数化序列实行规范化；Distance-求两点间距离；InitK-确定福利参数化的修正系数m_k；InitSita-计算弦线夹角的外角。

```
void InitU()
{
    if(m_aU.GetSize()>0) m_aU.RemoveAll();
    int i,n;
    n=m_aVertex.GetSize();
    m_aU.Add(0.0);
    switch(m_pType)
    {
        case 0: //规范均匀参数化
            for(i=1;i<n;i++) m_aU.Add(i);
            Normalize();
            break;
        case 1: //规范弦长参数化
            for(i=1;i<n;i++)
            {
                double temp;
                temp=Distance(m_aVertex[i-1],m_aVertex[i]);
                temp+=m_aU[i-1];
                m_aU.Add(temp);
            }
            Normalize();
            break;
        case 2: //规范向心参数化
            for(i=1;i<n;i++)
            {
                double temp;
                temp=sqrt(Distance(m_aVertex[i-1],m_aVertex[i]));
                temp+=m_aU[i-1];
                m_aU.Add(temp);
            }
            Normalize();
            break;
        case 3: //规范福利参数化
            InitK();
            for(i=1;i<n;i++)
            {
                double temp;
                temp=m_k[i]*Distance(m_aVertex[i-1],m_aVertex[i]);
                temp+=m_aU[i-1];
                m_aU.Add(temp);
            }
            Normalize();
            break;
        default:;
    }
}

void Normalize() //规范化
{
    int n=GetVertexCount();
    for(int i=0;i<n;i++) m_aU[i]=m_aU[i]/m_aU[n-1];
}

void InitK() //初始化修正系数
{
    if(m_k.GetSize()>0) m_k.RemoveAll();
    double tempk,temp1,temp2,temp3;
    InitSita();
    m_k.Add(0.0);
    for(int i=1;i<GetVertexCount();i++)
    {
        if(i==1) temp1=0.0;
        else temp1=Distance(m_aVertex[i-2],m_aVertex[i-1]);
        if(i==GetVertexCount()-1) temp3=0;
        else temp3=Distance(m_aVertex[i],m_aVertex[i+1]);
        temp2=Distance(m_aVertex[i-1],m_aVertex[i]);
        tempk=1+1.5*((temp1*m_sita[i-1])/(temp1+temp2)+
(temp3*m_sita[i])/(temp2+temp3));
        m_k.Add(tempk);
    }
}
```

```
}  
  
void InitSita() //计算弦线夹角的外角  
{  
    if(m_sita.GetSize()>0) m_sita.RemoveAll();  
    double temp,temp1,temp2,temp3;  
    m_sita.Add(0.0);  
    for(int i=1;i<GetVertexCount()-1;i++)  
    {  
        temp1=Distance(m_aVertex[i-1],m_aVertex[i]);  
        temp2=Distance(m_aVertex[i],m_aVertex[i+1]);  
        temp3=Distance(m_aVertex[i-1],m_aVertex[i+1]);  
        temp=acos((temp1*temp1+temp2*temp2-temp3*temp3)/  
            (2.0*temp1*temp2));  
        if((PI-temp)<(PI/2.0)) m_sita.Add(PI-temp);  
        else m_sita.Add(PI/2.0);  
    }  
    m_sita.Add(0.0);  
}
```